Pickling & Unpickling in Python

Pickling & Unpickling

- **Pickling:** pickling is a process where a python object hierarchy is converted into a byte stream.
  Unpickling: unpickling is the inverse of Pickling process where a byte stream is converted into an object hierarchy.
- ** Pickling and unpickling is alternatively known as **serialization**, **marshalling**, or **flattening**.

dump() and load() functions

- dump() and load() functions of pickle module perform pickling and unpickling Python data. The dump() function writes pickled object to a file and load() function unpickles data from file to Python object.

PICKLE MODULE

pickle.dump()

Used to write the object in a file

Syntax: pickle.dump(<structure>,file object)

Here, structure can be any sequence such as list, dictionary of Python.

And file object is the file handle of the file, in which to write.

pickle.load() Used to read the data from a file.

Syntax: Structure=pickle.load(file object)

Here, structure can be any sequence such as list, dictionary of Python.

And file object is the file handle of the file, from which to read.

```
importpickle
output_file=open("d:\\a.bin","wb")
myint=42
mystring="Python.mykvs.in!"
mylist=["python","sql","mysql"]
mydict={"name":"ABC","job":"XYZ"}
pickle.dump(myint,output_file)
pickle.dump(mystring,output_file)
pickle.dump(mylist,output_file)
pickle.dump(mydict,output_file)
output_file.close()
input_file=open("d:\\a.bin","rb")
myint=pickle.load(input_file)
mystring=pickle.load(input_file)
mylist=pickle.load(input_file)
mydict=pickle.load(input_file)
print("myint=%s"%myint)
print("mystring=%s"%mystring)
print("mylist=%s"%mylist)
print("mydict=%s"%mydict)
input_file.close()
```

Binary file R/W Operation using pickle module

Read records from a Binary file

```
f = open('d:/student.dat','rb')
while True:
try:
rec = pickle.load(f)
print('Roll Num:',rec['Rollno'])
print('Name:',rec['Name'])
print('Marks:',rec['Marks'])
except EOFError:
break
f.close()
```

THANK YOU!!!